

## 5.211 k\_used\_by

	DESCRIPTION	LINKS	GRAPH
<b>Origin</b>	Derived from <a href="#">used_by</a>		
<b>Constraint</b>	<code>k_used_by(SETS)</code>		
<b>Type</b>	<code>VARIABLES : collection(var-dvar)</code>		
<b>Argument</b>	<code>SETS : collection(set - VARIABLES)</code>		
<b>Restrictions</b>	<code>required(VARIABLES, var)</code> $ VARIABLES  \geq 1$ <code>required(SETS, set)</code> $ SETS  > 1$ <code>non_increasing_size(SETS, set)</code>		
<b>Purpose</b>	Given $ SETS $ sets of domain variables, the <code>k_used_by</code> constraint forces a <a href="#">used_by</a> constraint between each pair of consecutive sets.		
<b>Example</b>	$\left( \left\langle \begin{array}{l} \text{set} - \langle 1, 9, 1, 5, 2, 1 \rangle, \\ \text{set} - \langle 9, 1, 1, 1, 2, 5 \rangle, \\ \text{set} - \langle 1, 1, 2, 5 \rangle \end{array} \right\rangle \right)$ <p>The <code>k_used_by</code> constraint holds since:</p> <ul style="list-style-type: none"> <li>• The multiset of values <math>\{\{1, 1, 1, 2, 5, 9\}\}</math> associated with the second collection of variables is included into the multiset <math>\{\{1, 1, 1, 2, 5, 9\}\}</math> associated with the first collection of variables.</li> <li>• The multiset of values <math>\{\{1, 1, 2, 5\}\}</math> associated with the third collection of variables is included into the multiset <math>\{\{1, 1, 1, 2, 5, 9\}\}</math> associated with the second collection of variables.</li> </ul>		
<b>Typical</b>	$ VARIABLES  > 1$		
<b>Symmetries</b>	<ul style="list-style-type: none"> <li>• Items of <code>SETS</code> are <a href="#">permutable</a>.</li> <li>• Items of <code>SETS.set</code> are <a href="#">permutable</a>.</li> <li>• All occurrences of two distinct values of <code>SETS.set.var</code> can be <a href="#">swapped</a>; all occurrences of a value of <code>SETS.set.var</code> can be <a href="#">renamed</a> to any unused value.</li> </ul>		
<b>Arg. properties</b>	<a href="#">Contractible</a> wrt. <code>SETS</code> .		
<b>Remark</b>	Similarly to the <a href="#">k_same</a> constraint [151], finding out whether the <code>k_used_by</code> constraint has a solution or not is NP-hard when we have more than one <a href="#">used_by</a> constraint.		

**See also**

**common keyword:** `k_used_by_interval`, `k_used_by_modulo`,  
`k_used_by_partition` (*system of constraints*).  
**implied by:** `k_same`.  
**part of system of constraints:** `used_by`.  
**used in graph description:** `used_by`.

**Keywords**

**characteristic of a constraint:** sort based reformulation.  
**combinatorial object:** multiset.  
**constraint type:** system of constraints, decomposition.  
**modelling:** inclusion.

<b>Arc input(s)</b>	SETS
<b>Arc generator</b>	$\text{PATH} \mapsto \text{collection}(\text{set1}, \text{set2})$
<b>Arc arity</b>	2
<b>Arc constraint(s)</b>	$\text{used\_by}(\text{set1.set}, \text{set2.set})$
<b>Graph property(ies)</b>	$\text{NARC} =  \text{SETS}  - 1$

**Graph model**

Parts (A) and (B) of Figure 5.465 respectively show the initial and final graph associated with the **Example** slot. To each vertex corresponds a collection of variables, while to each arc corresponds a  $\text{used\_by}$  constraint.

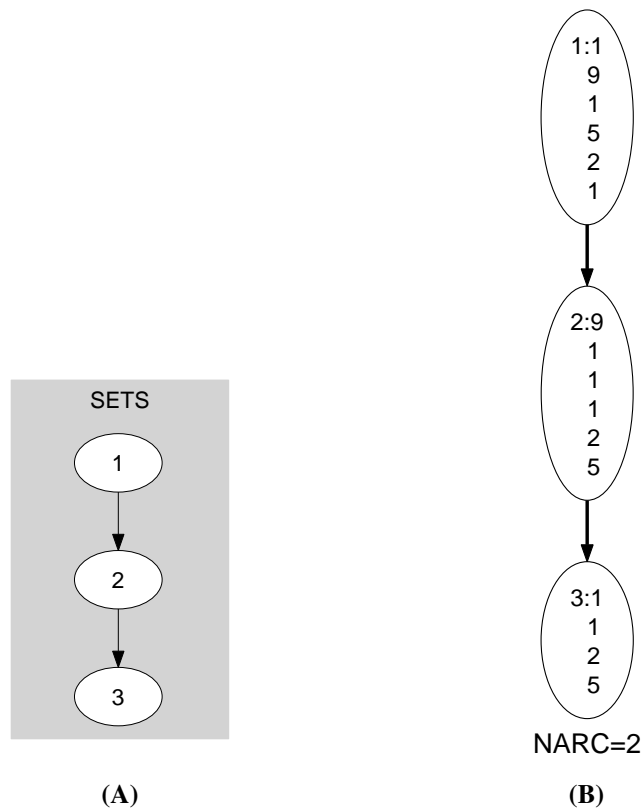


Figure 5.465: Initial and final graph of the  $k\text{-used\_by}$  constraint

20050814

1487